Progetto Intelligenza Artificiale
Pietro Zarri
30 agosto 2019

Alberi di decisione con dati mancanti

Introduzione:

Questo esercizio prevede il testing dell'algoritmo di classificazione tramite alberi di decisone. È anche implementata una tecnica per la gestione di eventuali dati mancanti. Il test è effettuato più volte al variare della probabilità di trovare un valore mancante all'interno del data set esaminato.

Teoria:

L'utilizzo degli alberi di decisione è una delle forme di apprendimento più semplice, ma allo stesso tempo di maggior successo ed è anche uno dei tipi di apprendimento induttivo più facili da implementare. Un albero di decisione prende in ingresso una situazione descritta da un insieme di attributi e restituisce un valore predetto di uscita per tale input. Il suo funzionamento consiste nell'eseguire in sequenza una serie di test. Ogni nodo corrisponde a un test su una delle proprietà e le diramazioni uscenti dal nodo indicano i possibili risultati. Le foglie rappresentano tutti i possibile valore predetti in uscita. Una volta allenato l'albero su un insieme, detto training set sarà possibile sfruttarlo per classificare situazioni di cui non conosciamo la classificazione.

Esperimento:

L'esercizio prevede in prima fase di implementare alberi di decisione e il loro algoritmo di apprendimento. Successivamente è necessario scegliere un algoritmo per la gestione di dati mancanti all'interno del data set. Scegliamo di

assegnare all'attributo mancante il valore più comune che assume rispetto a tutti gli altri esempi in quel nodo. Infine si valuta l'accuratezza, tramite *10-fold-cross-validation,* su ciascun data set al variare della probabilità di rimozione di valori in esso. I test sono effettuati su tre data sets diversi.

Realizzazione:

Per la realizzazione dell'esercizio è stato stato utilizzato il linguaggio Python ed è stata sfruttata la libreria *mathplotlib* per realizzare un grafico riassuntivo dell'esperimento.

I tre data sets, reperiti sul sito UCI Machine Learning Repository, sono:

- Car Evaluation Data Set
- Tic-Tac-Toe Endgame Data Set
- Qualitative Bankruptcy Data Set

Il progetto si compone di 7 file Python:

- *DecisonFork.py*: Rappresenta il nodo di un alberi di decisione: indica l'attributo su cui è eseguito il test e i possibili rami di uscita da tale nodo.
- *DecisionLeaf.py*: Rappresenta una foglia di un albero di decisione e quindi indica il risultato della classificazione a cui si è giunti.
- *DecisionTreeLearner.py*: Implementa l'algoritmo di apprendimento per alberi di decisione. Sono incluse anche le funzioni che permettono di scegliere l'attributo migliore su cui eseguire i test.
- *utils.py*: Contiene funzioni di supporto necessarie per l'algoritmo di apprendimento.
- *Dataset.py*: Questa classe serve per rappresentare il data set su cui cui viene modellato l'alberi di decisione. Contiene un insieme di esempi ciascuno con valori diversi sui vari attributi. Uno di tali attributi viene

scelto come “target” ed è quello che l’algoritmo proverà a predire. Sono presenti anche la funzione per la rimozioni in modo casuale e uniforme con probabilità fissata di valori e quella per implementare la strategia di rimpiazzo dei valori mancanti.

- *CrossValidation.py*: Implementa la tecnica della *cross-validation*. In particolare la *10-fold-cross validation* suddivide il dataset in 10 parti di ugual dimensione. Si esclude poi, in modo iterativo, un gruppo alla volta (*testing set*) e lo si cerca di predire con i gruppi non esclusi(*training set*). Ciò al fine di verificare la bontà del modello di predizione utilizzato.

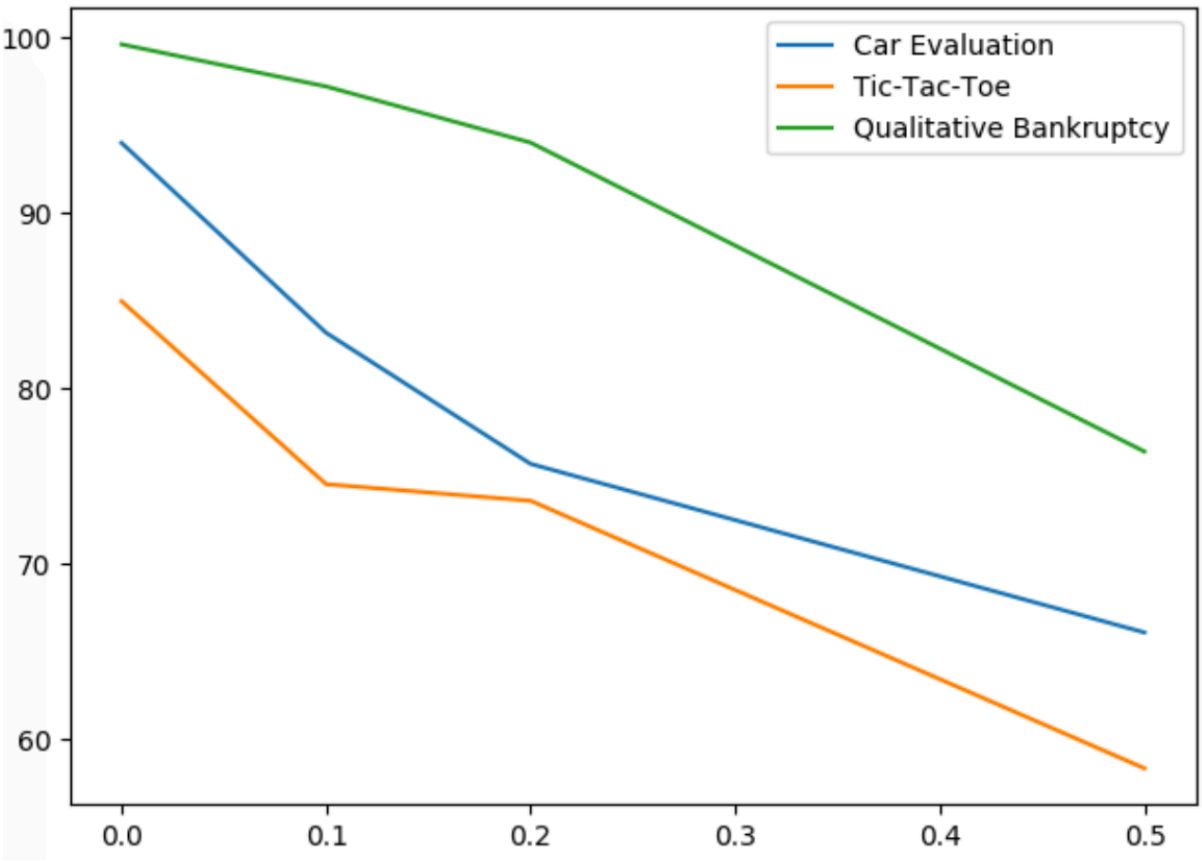
- *test.py*: In questo file prima è definita la funzione che esegue i test, in seguito, all’interno del main, sono creati ed esaminati i tre data sets. Infine è generato un grafico di riepilogo.

Risultati:

I risultati ottenuti dai test sui tre data sets, misurati tramite *10-fold-cross validation,* ovvero numero di classificazioni sbagliate diviso numero di classificazioni eseguite correttamente in percentuale, sono riassunti nella tabella sottostante. I valori indicano la percentuale di errori nella classificazione dei testing set quando la probabilità di rimozione di un valore all’interno del data set è $0, 0.1, 0.2, 0.5$.

	0	0.1	0.2	0.5
Car Evalutation	6.0173%	16.8366%	24.3067%	33.9111%
Tic-Tac-toe	15.0362%	25.4627%	26.4079%	41.6502%
Qualitative Bankruptcy	0.4%	2.8%	6.0%	23.6%

Di seguito è anche riportato un grafico che rappresenta invece la percentuale di valori correttamente predetti.



Conclusione:

Come possiamo vedere, sia dalla tabella che dal grafico, le prestazioni di classificazione da parte degli alberi di decisioni vanno a diminuire in tutti e tre i data sets all'aumentare della quantità di dati mancanti in essi. Possiamo però osservare che grazie alla politica adottata per rimpiazzare i buchi creati nei data sets si cerca di mantenere un'efficienza accettabile. In particolare il data set *"Quality Bankruptcy"* anche con un 50% di dati rimossi mantiene un accuratezza di classificazione di più del 75%.